**I. Course Details** - ELEE 390 B, Developing Concurrent Software, Spring 2014,

This is an elective course for Electrical and Computer Engineers. Note this is a 1 credit elective and will only run for a third of the semester (5 evening classes plus one class for the final).

**Catalog Description** - This course focuses on the principles of design and implementation of software which uses concurrency to take advantage of modern process architecture to improve performance and scalability.

**Prerequisites** - COMP 220

**Course Overview** - This is a seminar course which will review the building blocks of concurrent software development, including mutexes, atomics, asynchronous operations, threads, tasks, and MPI. Students will learn the test driven design software development technique, and will gain experience with the Git source control tool. Students will be required to write software both in and out of class, and will be encouraged to collaborate on some assignments.

**Meeting** - Mondays, 6:30-9 PM, Jan. 20 - Feb. 24, HAL room 209

**Course Web Page** - <https://github.com/joshpeterson/DevelopingConcurrentSoftware>

**Text** - C++ Concurrency in Action - Anthony Williams (optional, but recommended)

**Faculty Details** - Josh Peterson, [petersonjm1@gmail.com](mailto:petersonjm1@gmail.com), 724-825-8881

**Office Hours** - before and after class

**II. ABET Student Outcomes** - By the conclusion of the course, students must demonstrate:

(b) An ability to design and conduct experiments as well as to analyze and interpret data.

Test driven design serves as an application of the the scientific method to software development.

(d) An ability to function on multidisciplinary teams.

Working collaboratively on development assignments will require students to apply the knowledge and skills of each member of the team to achieve the goal.

(k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

Use of modern multi-processor hardware will provide students with experience necessary to scale software applications for future architectures.

**III. Course Outcomes** - Upon completion of the course, students will:

1. Apply test driven design as a development technique.
2. **Assessment** - class discussion, collaborative homework problems, test questions
3. Use the Git source control system for committing, reverting, and branching.
4. **Assessment** - class discussion, collaborative homework questions, test questions
5. Understand and employ the tools necessary for concurrency, including mutexes, atomics, and threads.
6. **Assessment** - class discussion, collaborative homework questions, test questions
7. Understand and employ patterns for concurrency, including asynchronous operations, producer/consumer queues, tasks, and MPI.
8. **Assessment** - class project, collaborative homework questions, test questions

**IV. Methodology**

**Teaching Methods** - This course will include lectures, class discussions and, in-class group and pair programming practice.

**Assessment Procedures** - There will be five homework assignments, a quiz and a final exam, and a brief paper. I also expect each student to participate fully in all class exercises and discussions.

**Grading** -

50% Homework (10% for each assignment)

10% Attendance and Participation

10% Quiz

10% Paper

30% Final Exam

Scale - 100-98 A+, 97-92 A, 91-90 A-, 89-88 B+, 87-82 B, 81-80 B-, etc.

**V. Policies**

**Attendance** - Per school policy, 1 unexcused absence is permitted. Additional absences will detract from the attendance portion of the final grade.

**Academic Integrity** -

The practice of good Christian ethical behavior is essential for maintaining good order in the classroom, providing an enriching learning experience for students and as training as a practicing engineering professional upon graduation. Moreover, the practice of Christian ethics in all aspects of one’s life leads to good moral and spiritual development. This practice is manifested in the College’s Academic Integrity policy.

Students are expected to adhere to the "Honesty in Learning" policy as outlined in the *Bulletin* and *Crimson*. Violations will be dealt with as outlined therein.

Application to this course - Discussion with other students of class material and of background material relevant to homework problems is encouraged. Each homework paper submitted should represent *only* your work, however. You may collaborate with other students only while conceptualizing the problem and while outlining the solution strategy. You should not copy homework solutions from other students.

Final Exam Policy -

As stated in the Bulletin, final exams must be administered according to the time scheduled by the Registrar’s office, and cannot be changed to suit the convenience of the student. It is your responsibility to schedule your travel and work plans accordingly. Students with a Provost’s excused absence will receive permission to reschedule their final exam. In addition, any student who has three exams scheduled on a given day may request permission from the Dean of the school of the student first major to reschedule one exam; however, the Dean is not required to grant such a request. For this course, the final exam has been scheduled for the final meeting day of the course, February 24, 2014.

**Computer Use Policy** -

Student will be required to use their computers for a number of in-class activities. You are welcome to use their computers for note-taking as well. The manner in which you use your computer in class is considered a matter of honor and professionalism. Inappropriate use of a computer in the classroom may be viewed as being disrespectful to the instructor, is often distracting to other students, and is unprofessional.

Judgment as to the appropriateness of student computer use is at the discretion of the instructor. The consequences for violating this policy are also at the discretion of the instructor.

**Homework Policy** - Homework and writing assignments are due at the beginning of class on the due date. Assignments turned in after that but before midnight (of that day) receive a 10% penalty. Assignments turned in by midnight of the following day receive a 30% penalty. After midnight the following day, the assignment receives 0%.

**Test Policy** - Tests/exams are closed book. You are allowed to bring in 1 page (8.5x11") of notes. I may review these note sheets during or after the test. Unless otherwise stated, no calculators are allowed, there will be little or no math on the tests. The final will be cumulative.

**VI. Schedule and Readings**

Tentative Schedule:

|  |  |  |
| --- | --- | --- |
| **Date** | **Topics** | **Suggested Reading** |
| Jan. 20 | Intro to Git, CPU architecture, test driven design, Dennard scaling | Williams, chapter 1 |
| Jan. 27 | Tools for concurrency, threads, mutexes, asynchronous operations | Williams, chapters 2, 3, 4 |
| Feb. 3 | Atomics and the C++ memory model, designing concurrent software, task-based concurrency | Williams, chapter 5 (up to section 5.3.2), chapter 8 (up to section 8.4) |
| Feb. 10 | Case studies: atomics and object lifetime, making a cache thread safe, synchronizing a thread pool | None |
| Feb. 17 | Introduction to MPI, Hybrid parallelism | None |
| Feb. 24 | Final exam | None |